Assignment 8

Don Johnson
donj@bu.edu

1.2 Technical Task

All results are in the zip file in a directory call "Data"

1. Five frames each of "video", "magnitude" and "angle" images for the optical flow assignment.
2. Two AVI files, "background_source.avi" and "background_result.avi" for the background subtraction assignment.

Source code additions and modifications have been labeled with "//Modified by ..."

1.3 Lecture Preparation

1. n-space linear programming is a generalization of 2-space LP. 2-dimension LP defines a linear function to be maximized:

$$f(x_1, x_2) = c_1 x_1 + c_2 x_2$$

and a set of constraints,

$$a_{11} x_1 + a_{12} x_2 \leq b_1$$
$$a_{21} x_1 + a_{22} x_2 \leq b_2$$
$$a_{31} x_1 + a_{32} x_2 \leq b_3$$
$$\cdots$$

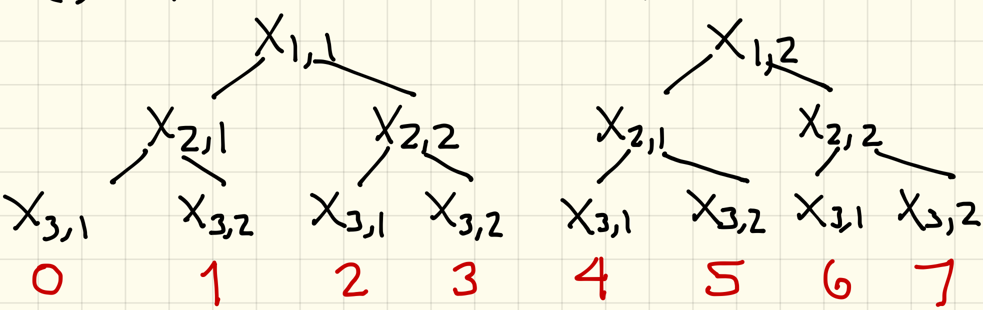
and $x_1 \geq 0$, $x_2 \geq 0$

which when graphed yield a convex polygon. The vertices of the polygon are substuted back into $f(x_1, x_2)$ to see which values of x_1, x_2 result in a maximal value.

2. Integer Programming is a special case of LP when all variablex X_n must be integers.

3. Detections (X_n, X_m) where $n = 1, 2, 3, \ldots$ representing points in time $t_1, t_2, t_3 \ldots$ and where m is a dection ID. For example, it could be a pointer to the (x, y) centroid of a unique object. Next, we assign a column number in red for each of the possible leaves/paths at the bottom.

(a) Tree 1 Tree 2



Column Positions

(b) Matrix Representation

$$[A] \quad \begin{array}{c|c}
 & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\
\begin{matrix} X_{1,1} \\ X_{1,2} \\ X_{2,1} \\ X_{2,2} \\ X_{3,1} \\ X_{3,2} \end{matrix} &
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\end{array}$$

← Leaf Number

↑ Detection

* All possible paths are represented in the matrix

(c) Two Sample Leaves and Corresponding Paths

Leaf 0

$X_{1,1} \to X_{2,1} \to X_{3,1}$

$X_0 = 101010$

Leaf 5

$X_{1,2} \to X_{2,1} \to X_{3,2}$

$X_5 = 011001$

Test for overlapping paths on next page →

(c) continued

To test if path n overlaps with path m, take the dot product of X_n with X_m. If the dot product equals zero, the paths do not overlap. Dot products greater than zero mean the paths do overlap.

$X_0 = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$
$X_5 = [0\ 1\ 1\ 0\ 0\ 1\ 1]$
$X_7 = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$

Column vectors from matrix A

$X_0 \cdot X_5 = 1$ so paths 0 and 5 overlap

$X_0 \cdot X_7 = 0$ so path 0 and 7 do not overlap

If you take $A^T X_n$, the result will be a column vector that has zero values on the mth row corresponding to a non-overlapping path m.

In this example, if you compute $A^T \cdot A$, the are zeros in cells a_{07}, a_{16}, a_{25}, a_{34}, a_{43}, a_{52}, a_{61} and a_{70}

The subscripts of those cells indicate the non-overlapping paths. See MATLAB demonstration on the next page →

```
>> A
A =

     1     0     1     0     1     0     1     0
     0     1     0     1     0     1     0     1
     1     1     0     0     1     1     0     0
     0     0     1     1     0     0     1     1
     1     1     1     1     0     0     0     0
     0     0     0     0     1     1     1     1

>> transpose(A)
ans =

     1     0     1     0     1     0
     0     1     1     0     1     0
     1     0     0     1     1     0
     0     1     0     1     1     0
     1     0     1     0     0     1
     0     1     1     0     0     1
     1     0     0     1     0     1
     0     1     0     1     0     1

>> transpose(A)*A(:,1)
ans =

     3
     2
     2
     1
     2
     1
     1
     0

>> transpose(A)*A
ans =

     3     2     2     1     2     1     1     0
     2     3     1     2     1     2     0     1
     2     1     3     2     1     0     2     1
     1     2     2     3     0     1     1     2
     2     1     1     0     3     2     2     1
     1     2     0     1     2     3     1     2
     1     0     2     1     2     1     3     2
     0     1     1     2     1     2     2     3
```